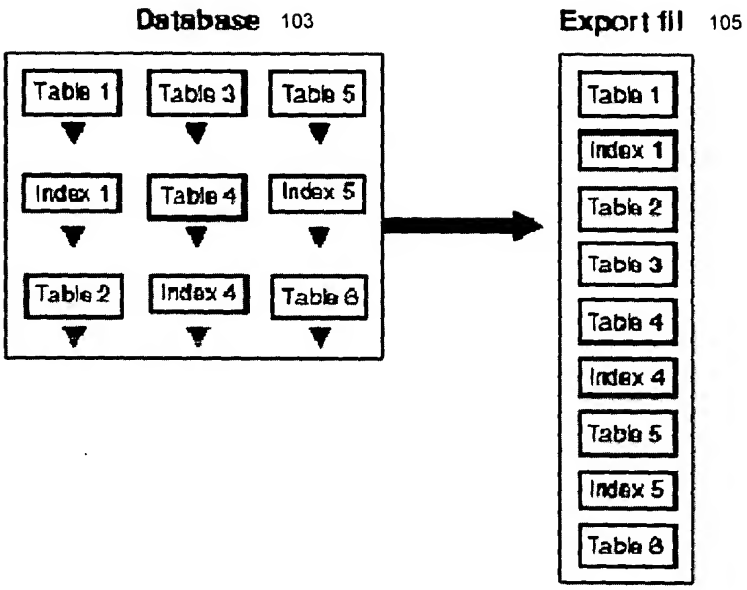
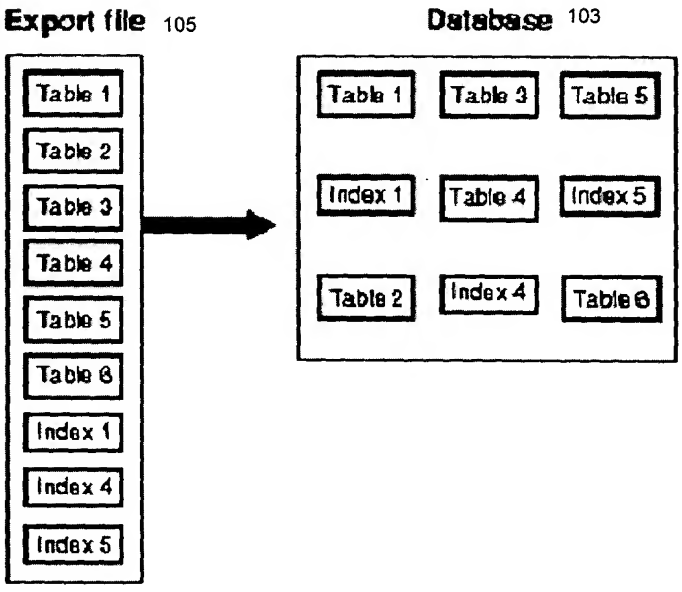


export 101



import 107

Fig. 1 PRIOR ART

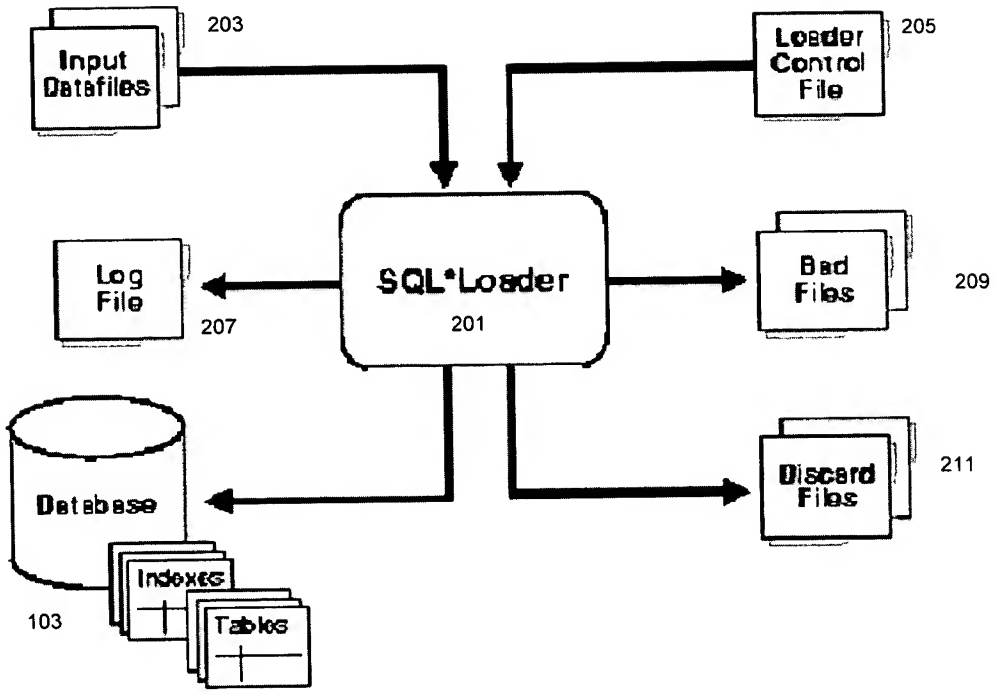


Fig. 2 PRIOR ART

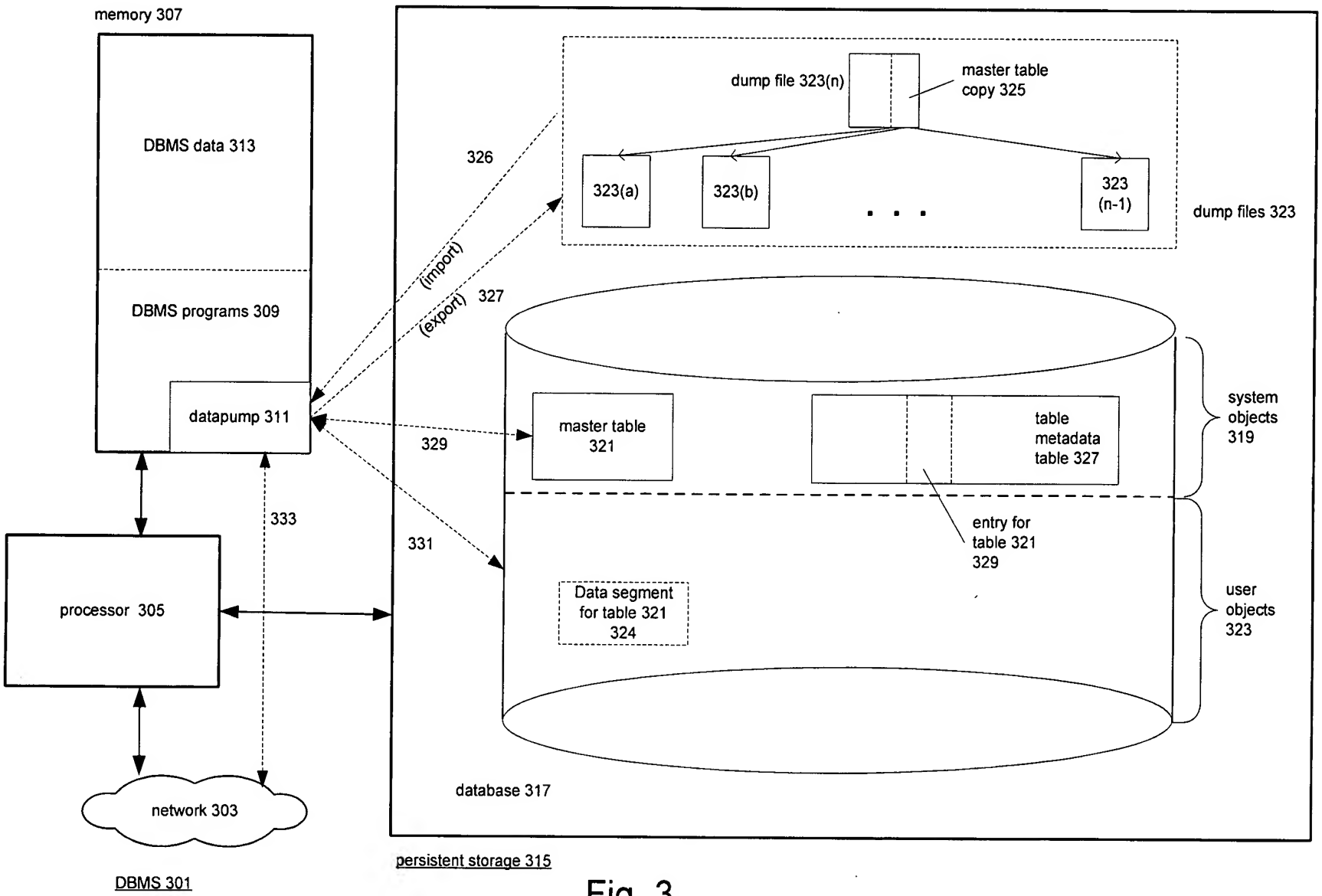

memory 307
DBMS data 313
DBMS programs 309
datapump 311
processor 305
network 303
DBMS 301
333
331
329
326
(import)
(export)
327
dump file 323(n)
master table copy 325
323(a)
323(b)
323 (n-1)
dump files 323
master table 321
table metadata table 327
system objects 319
entry for table 321 329
Data segment for table 321 324
user objects 323
database 317
persistent storage 315

Fig. 3

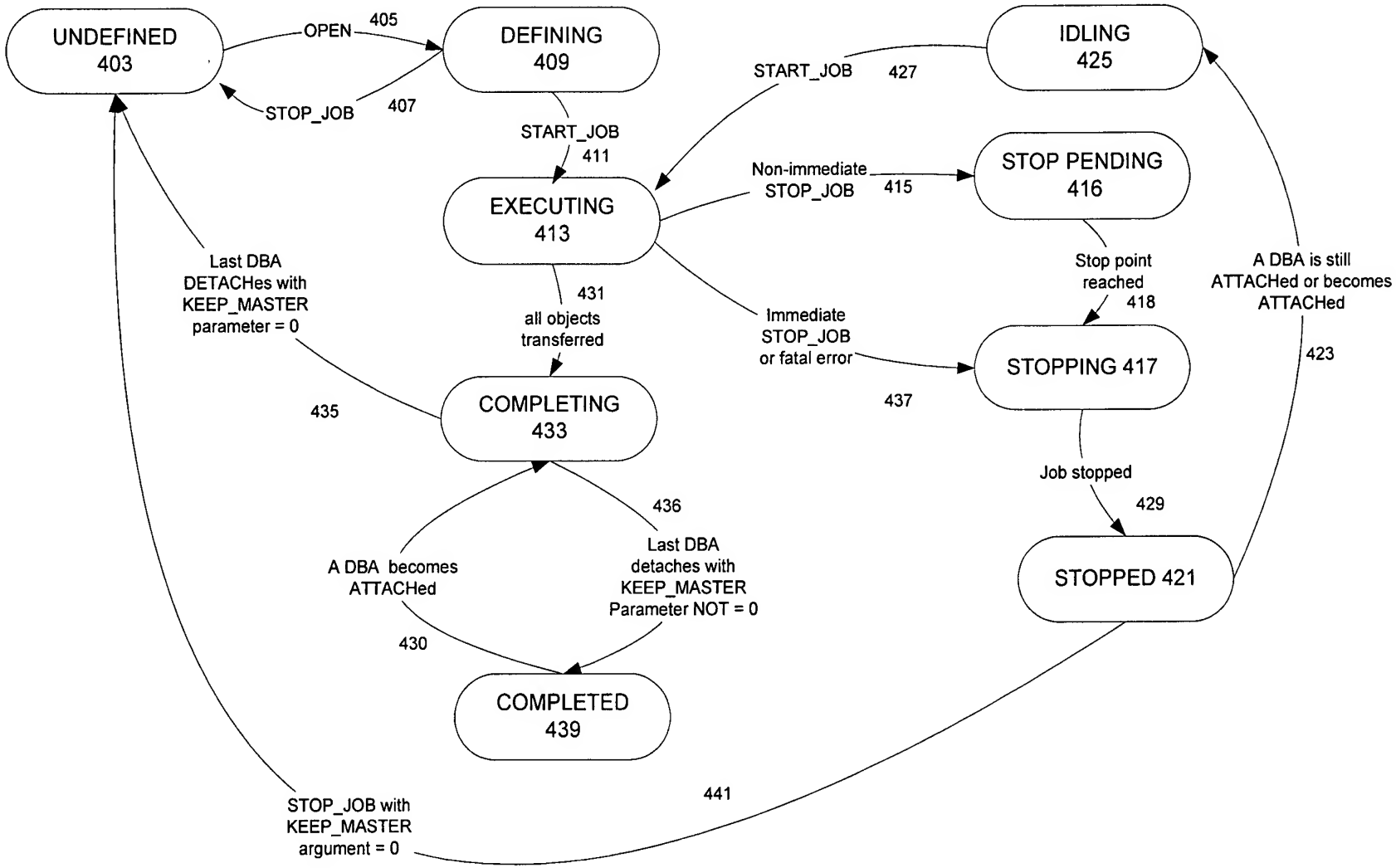

UNDEFINED
403
OPEN
405
DEFINING
409
STOP_JOB
407
START_JOB
411
EXECUTING
413
IDLING
425
START_JOB
427
Non-immediate
STOP_JOB
415
STOP PENDING
416
Last DBA
DETACHes with
KEEP_MASTER
parameter = 0
431
all objects
transferred
Immediate
STOP_JOB
or fatal error
Stop point
reached
418
A DBA is still
ATTACHed or becomes
ATTACHed
423
STOPPING 417
437
435
COMPLETING
433
Job stopped
429
436
A DBA becomes
ATTACHed
Last DBA
detaches with
KEEP_MASTER
Parameter NOT = 0
STOPPED 421
430
COMPLETED
439
441
STOP_JOB with
KEEP_MASTER
argument = 0
401

Fig. 4

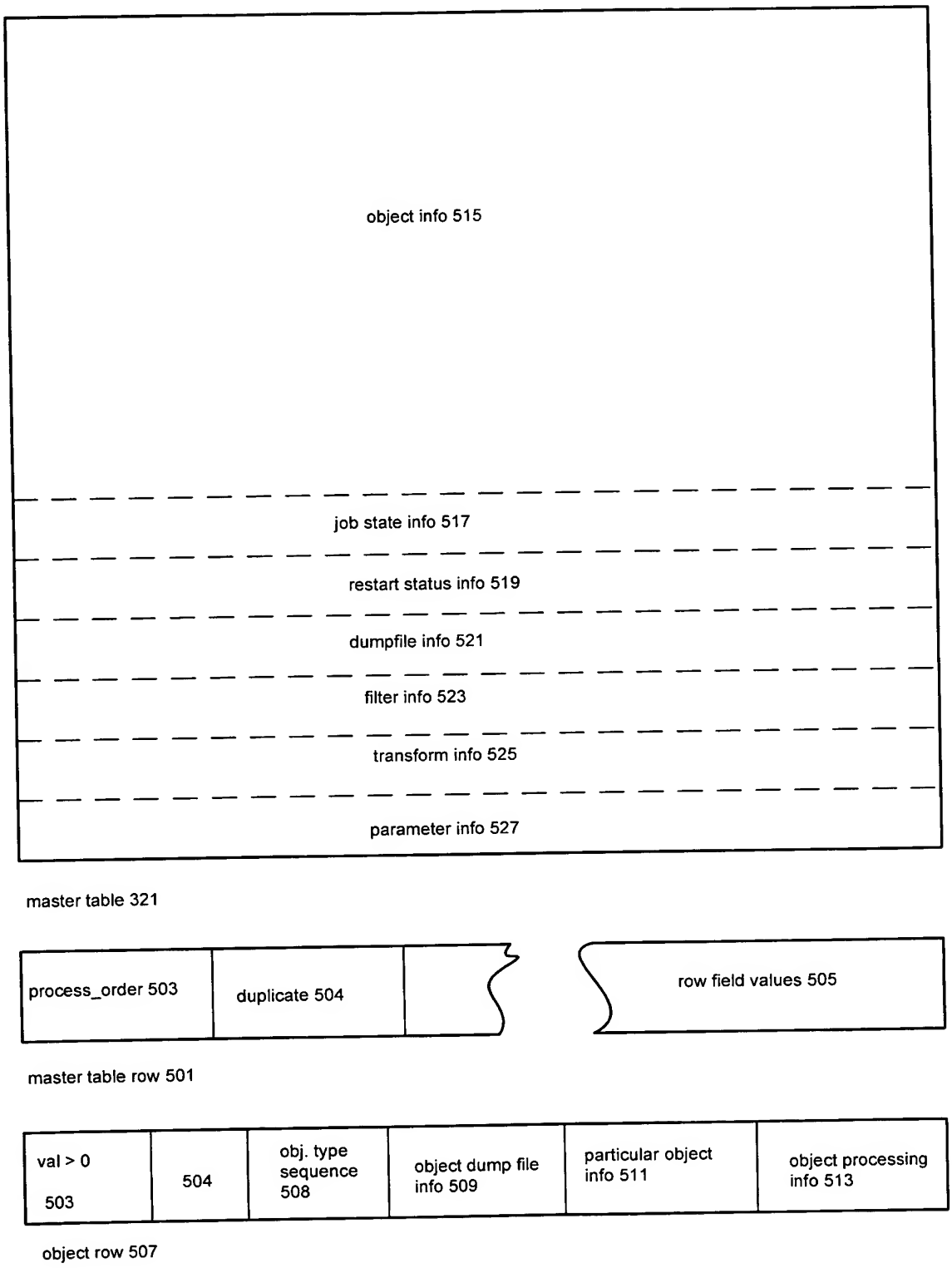

object info 515
job state info 517
restart status info 519
dumpfile info 521
filter info 523
transform info 525
parameter info 527
master table 321
process_order 503
duplicate 504
row field values 505
master table row 501
val > 0
503
504
obj. type sequence 508
object dump file info 509
particular object info 511
object processing info 513
object row 507

Fig. 5

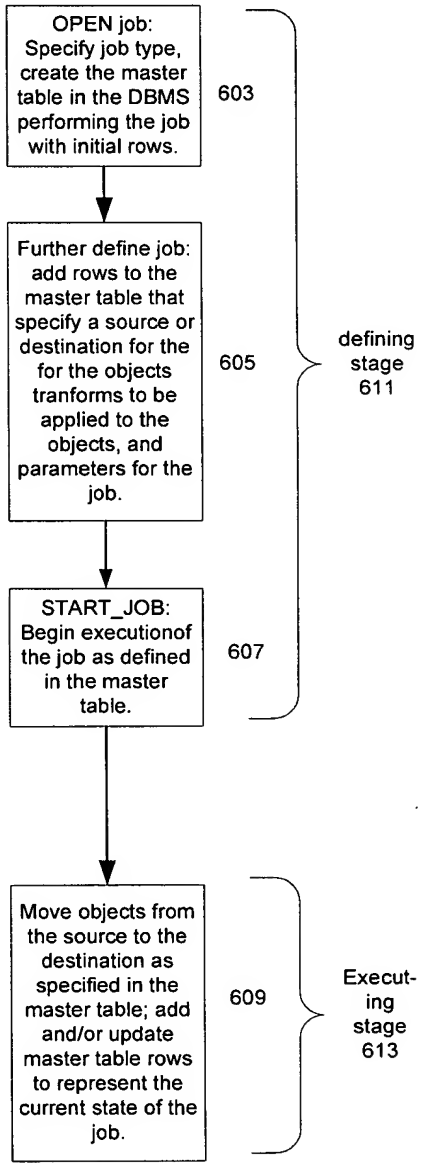

OPEN job: Specify job type, create the master table in the DBMS performing the job with initial rows.
603
Further define job: add rows to the master table that specify a source or destination for the for the objects tranforms to be applied to the objects, and parameters for the job.
605
START_JOB: Begin executionof the job as defined in the master table.
607
defining stage 611
Move objects from the source to the destination as specified in the master table; add and/or update master table rows to represent the current state of the job.
609
Execut-ing stage 613

Fig. 6

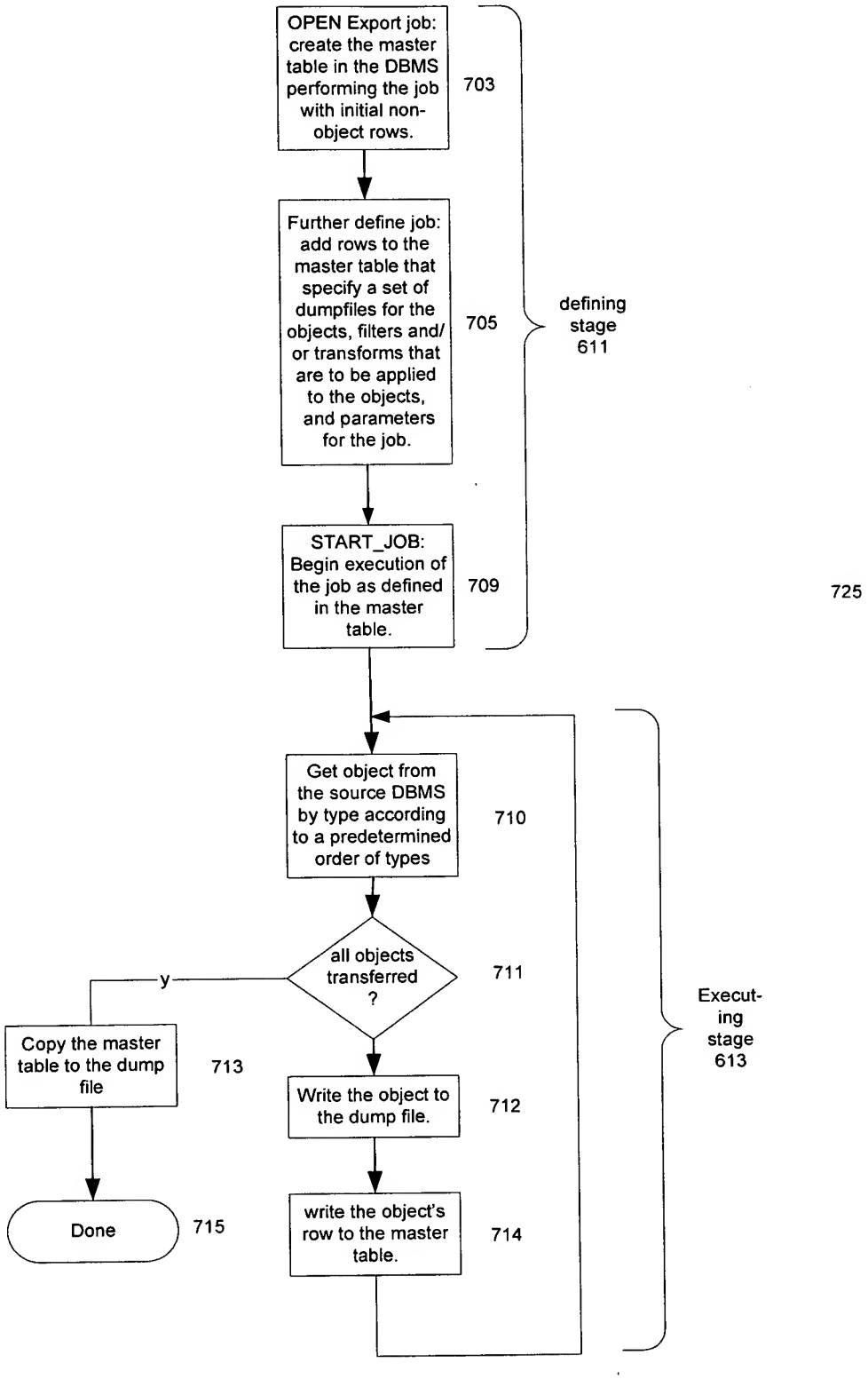

OPEN Export job: create the master table in the DBMS performing the job with initial non-object rows.
703
Further define job: add rows to the master table that specify a set of dumpfiles for the objects, filters and/or transforms that are to be applied to the objects, and parameters for the job.
705
START_JOB: Begin execution of the job as defined in the master table.
709
defining stage 611
725
Get object from the source DBMS by type according to a predetermined order of types
710
all objects transferred ?
711
y
Copy the master table to the dump file
713
Write the object to the dump file.
712
write the object's row to the master table.
714
Done
715
Executing stage 613
EXPORT 701

Fig. 7

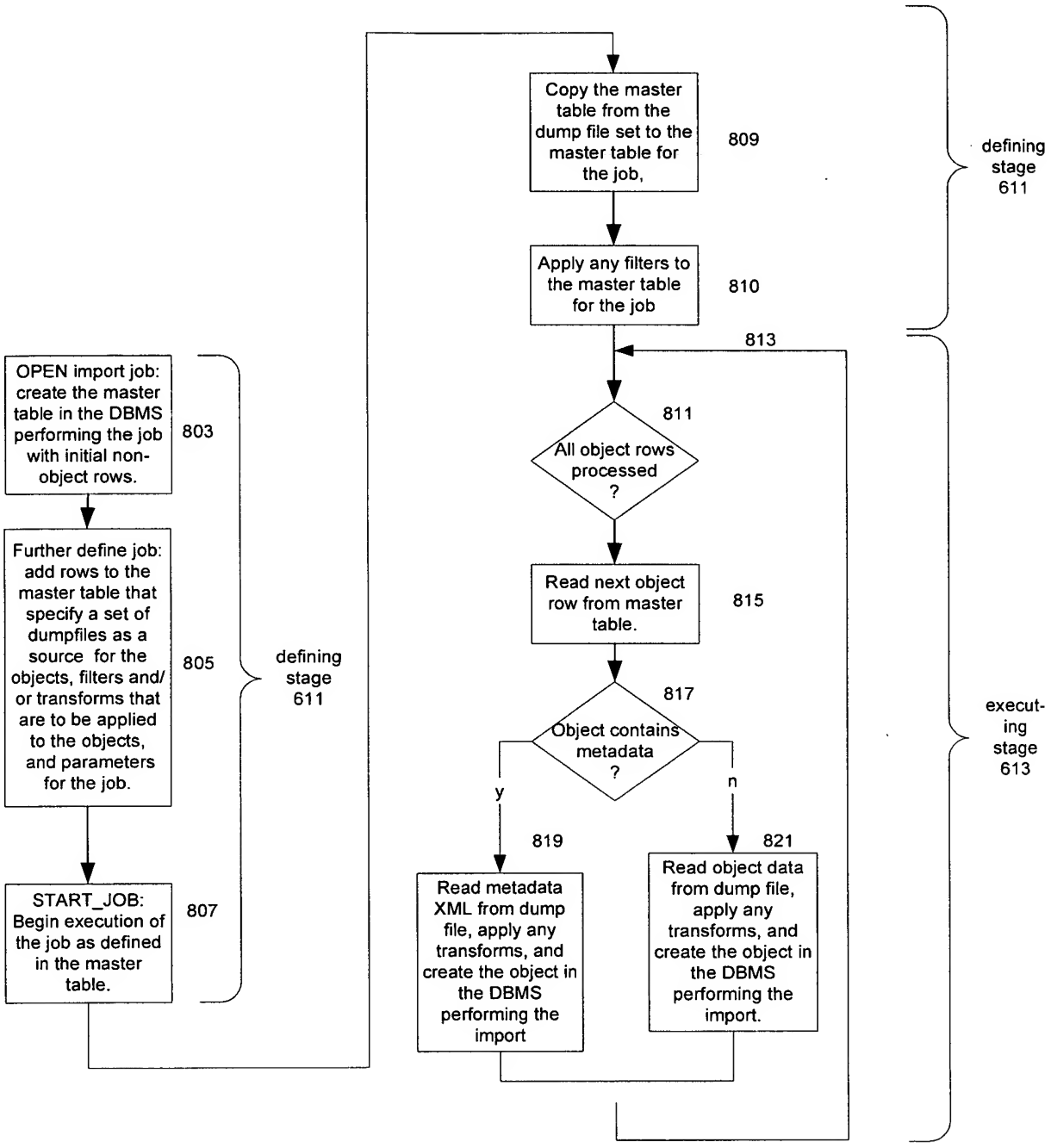

OPEN import job: create the master table in the DBMS performing the job with initial non-object rows.
803
Further define job: add rows to the master table that specify a set of dumpfiles as a source for the objects, filters and/or transforms that are to be applied to the objects, and parameters for the job.
805
START_JOB: Begin execution of the job as defined in the master table.
807
defining stage 611
Copy the master table from the dump file set to the master table for the job,
809
Apply any filters to the master table for the job
810
defining stage 611
813
All object rows processed ?
811
Read next object row from master table.
815
Object contains metadata ?
817
y
n
Read metadata XML from dump file, apply any transforms, and create the object in the DBMS performing the import
819
Read object data from dump file, apply any transforms, and create the object in the DBMS performing the import.
821
executing stage 613
IMPORT from dump file set 801

Fig. 8

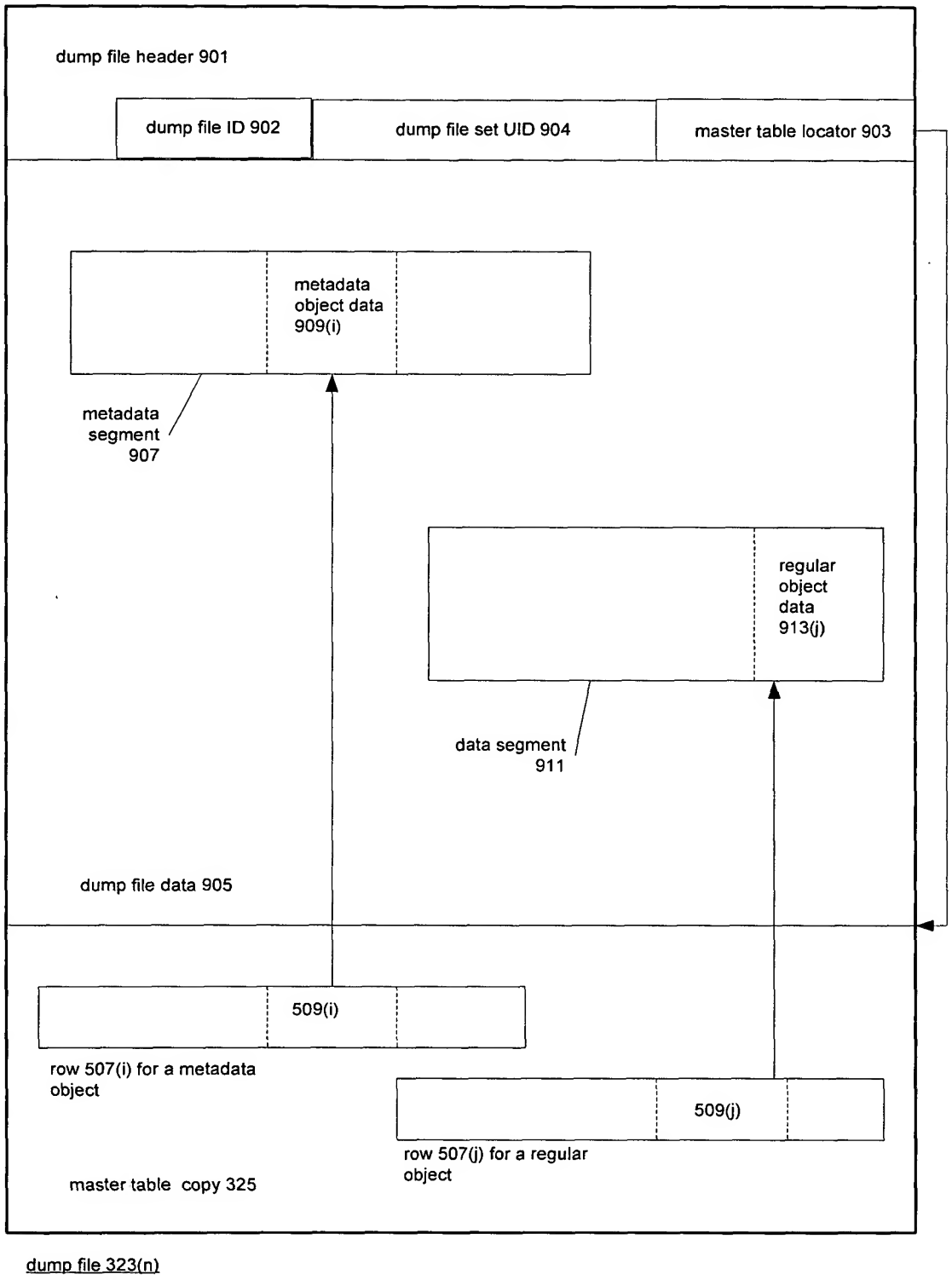
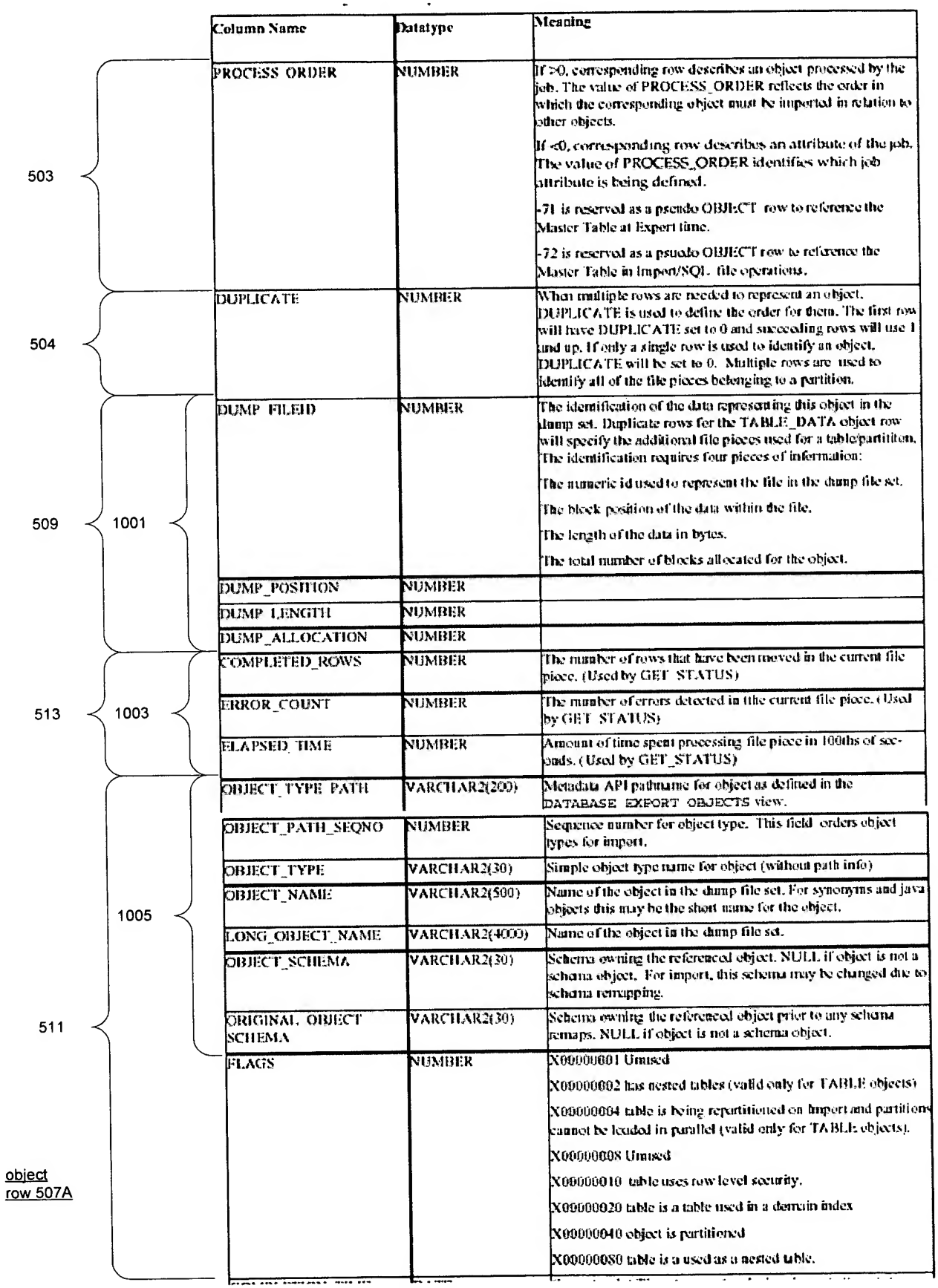
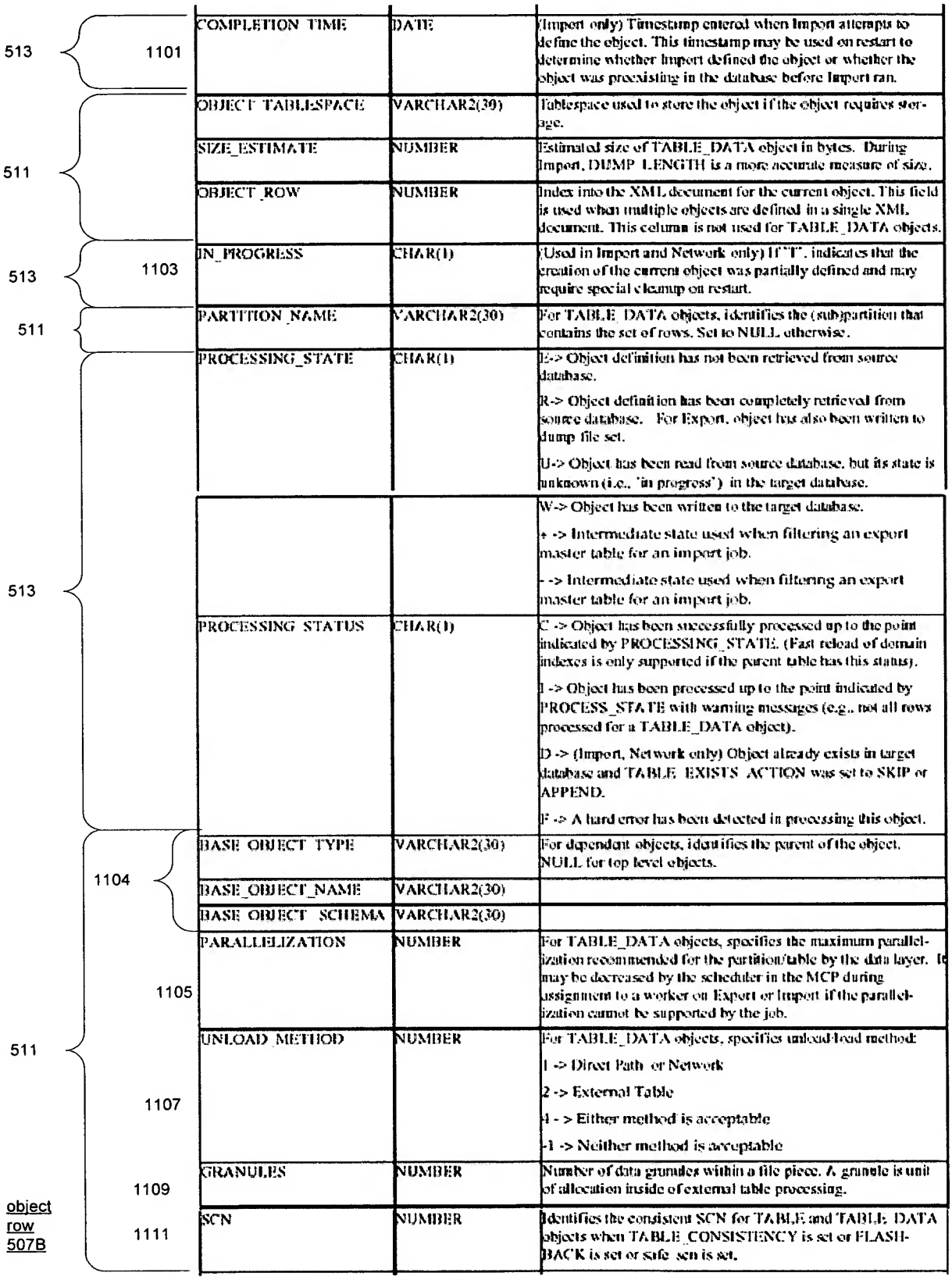

dump file header 901
dump file ID 902
dump file set UID 904
master table locator 903
metadata object data 909(i)
metadata segment 907
regular object data 913(j)
data segment 911
dump file data 905
509(i)
row 507(i) for a metadata object
509(j)
row 507(j) for a regular object
master table copy 325
dump file 323(n)

Fig. 9



		Column Name	Datatype	Meaning
503		PROCESS_ORDER	NUMBER	If >0, corresponding row describes an object processed by the job. The value of PROCESS_ORDER reflects the order in which the corresponding object must be imported in relation to other objects. If <0, corresponding row describes an attribute of the job. The value of PROCESS_ORDER identifies which job attribute is being defined. -71 is reserved as a pseudo OBJECT row to reference the Master Table at Export time. -72 is reserved as a pseudo OBJECT row to reference the Master Table in Import/SQL_file operations.
504		DUPLICATE	NUMBER	When multiple rows are needed to represent an object, DUPLICATE is used to define the order for them. The first row will have DUPLICATE set to 0 and succeeding rows will use 1 and up. If only a single row is used to identify an object, DUPLICATE will be set to 0. Multiple rows are used to identify all of the file pieces belonging to a partition.
509	1001	DUMP_FILEID	NUMBER	The identification of the data representing this object in the dump set. Duplicate rows for the TABLE_DATA object row will specify the additional file pieces used for a table/partition. The identification requires four pieces of information: The numeric id used to represent the file in the dump file set. The block position of the data within the file. The length of the data in bytes. The total number of blocks allocated for the object.
		DUMP_POSITION	NUMBER	
		DUMP_LENGTH	NUMBER	
		DUMP_ALLOCATION	NUMBER	
513	1003	COMPLETED_ROWS	NUMBER	The number of rows that have been moved in the current file piece. (Used by GET_STATUS)
		ERROR_COUNT	NUMBER	The number of errors detected in the current file piece. (Used by GET_STATUS)
		ELAPSED_TIME	NUMBER	Amount of time spent processing file piece in 100ths of seconds. (Used by GET_STATUS)
511	1005	OBJECT_TYPE_PATH	VARCHAR2(200)	Metadata API pathname for object as defined in the DATABASE_EXPORT_OBJECTS view.
		OBJECT_PATH_SEQNO	NUMBER	Sequence number for object type. This field orders object types for import.
		OBJECT_TYPE	VARCHAR2(30)	Simple object type name for object (without path info)
		OBJECT_NAME	VARCHAR2(500)	Name of the object in the dump file set. For synonyms and java objects this may be the short name for the object.
		LONG_OBJECT_NAME	VARCHAR2(4000)	Name of the object in the dump file set.
		OBJECT_SCHEMA	VARCHAR2(30)	Schema owning the referenced object. NULL if object is not a schema object. For import, this schema may be changed due to schema remapping.
		ORIGINAL_OBJECT_SCHEMA	VARCHAR2(30)	Schema owning the referenced object prior to any schema remaps. NULL if object is not a schema object.
object row 507A		FLAGS	NUMBER	X00000001 Unused X00000002 has nested tables (valid only for TABLE objects) X00000004 table is being repartitioned on import and partitions cannot be loaded in parallel (valid only for TABLE objects). X00000008 Unused X00000010 table uses row level security. X00000020 table is a table used in a domain index X00000040 object is partitioned X00000080 table is a used as a nested table.

Fig. 10



513 1101	COMPLETION_TIME	DATE	(Import only) Timestamp entered when Import attempts to define the object. This timestamp may be used on restart to determine whether Import defined the object or whether the object was preexisting in the database before Import ran.
511	OBJECT_TABLESPACE	VARCHAR2(30)	Tablespace used to store the object if the object requires storage.
	SIZE_ESTIMATE	NUMBER	Estimated size of TABLE_DATA object in bytes. During Import, DUMP_LENGTH is a more accurate measure of size.
	OBJECT_ROW	NUMBER	Index into the XML document for the current object. This field is used when multiple objects are defined in a single XML document. This column is not used for TABLE_DATA objects.
513 1103	IN_PROGRESS	CHAR(1)	(Used in Import and Network only) If 'I', indicates that the creation of the current object was partially defined and may require special cleanup on restart.
511	PARTITION_NAME	VARCHAR2(30)	For TABLE_DATA objects, identifies the (sub)partition that contains the set of rows. Set to NULL otherwise.
513	PROCESSING_STATE	CHAR(1)	E-> Object definition has not been retrieved from source database. R-> Object definition has been completely retrieval from source database. For Export, object has also been written to dump file set. U-> Object has been read from source database, but its state is unknown (i.e., 'in progress') in the target database. W-> Object has been written to the target database. + -> Intermediate state used when filtering an export master table for an import job. - -> Intermediate state used when filtering an export master table for an import job.
	PROCESSING_STATUS	CHAR(1)	C -> Object has been successfully processed up to the point indicated by PROCESSING_STATE. (Fast reload of domain indexes is only supported if the parent table has this status). I -> Object has been processed up to the point indicated by PROCESS_STATE with warning messages (e.g., not all rows processed for a TABLE_DATA object). D -> (Import, Network only) Object already exists in target database and TABLE_EXISTS_ACTION was set to SKIP or APPEND. F -> A hard error has been detected in processing this object.
1104	BASE_OBJECT_TYPE	VARCHAR2(30)	For dependant objects, identifies the parent of the object. NULL for top level objects.
	BASE_OBJECT_NAME	VARCHAR2(30)	
	BASE_OBJECT_SCHEMA	VARCHAR2(30)	
1105	PARALLELIZATION	NUMBER	For TABLE_DATA objects, specifies the maximum parallelization recommended for the partition/table by the data layer. It may be decreased by the scheduler in the MCP during assignment to a worker on Export or Import if the parallelization cannot be supported by the job.
511 1107	UNLOAD_METHOD	NUMBER	For TABLE_DATA objects, specifies unload/load method: 1 -> Direct Path or Network 2 -> External Table 4 - > Either method is acceptable -1 -> Neither method is acceptable
1109	GRANULES	NUMBER	Number of data granules within a file piece. A granule is unit of allocation inside of external table processing.
object row 507B 1111	SCN	NUMBER	Identifies the consistent SCN for TABLE and TABLE_DATA objects when TABLE_CONSISTENCY is set or FLASHBACK is set or safe_scn is set.

Fig. 11

511 (object row 507C)

1201 DOMAIN_INDEX	VARCHAR2(30)	If object is a secondary object, identifies the domain index that created the object. NULL otherwise.
DOMAIN_INDEX_SCHEMA	VARCHAR2(30)	
1202 GRANTOR	VARCHAR2(30)	For grants, the schema that originally created the grant.
1203 XML_CLOB	CLOB	For Table objects, the XML representation of the metadata to recreate the object. Also used to load data for Table_data objects within the table.

	Column Name	Datatype	Meaning
503	PROCESS_ORDER	NUMBER	-1 for Export jobs. -2 for Import and SQL_File jobs.
504	DUPLICATE	NUMBER	0
1207	OBJECT_NAME	VARCHAR2(500)	Name of the job (should be the same as the table name)
	OBJECT_LONG_NAME	VARCHAR2(4000)	Name of the job (should be the same as the table name)
	USER_NAME	VARCHAR2(30)	Username who initiated operation. (This should be same name as the owner of the Master Table.)
	OPERATION	VARCHAR2(30)	One of the following values: EXPORT, IMPORT, SQL_FILE.
	JOB_MODE	VARCHAR2(30)	One of the following values: FULL, SCHEMA, TABLE, TABLESPACE, TRANSPORTABLE
	REMOTE_LINK	VARCHAR2(4000)	Network link used for job (Null if none).
	VERSION	NUMBER	Version control for Master Table format.
	DB_VERSION	VARCHAR2(30)	The version of the database objects for this operation.
1213	STATE	VARCHAR2(30)	One of the following values: DEFINING, EXECUTING, IDLING, STOPPING, STOPPED, WAITING, COMPLETING, COMPLETED.
	PHASE	NUMBER	The sequence position in the tasks to accomplish the job. The meaning of each number is dependent upon the operation being performed.
	GUID	RAW(16)	Globally unique identifier for describing the job. The GUID is used to label files as belonging to the job.
1215	START_TIME	DATE	Datetime when the job was OPENed.
	BLOCK_SIZE	NUMBER	ksfq block size used in job. This is also defined in the header of each dump file.
	METADATA_BUFFER_SIZE	NUMBER	number of blocks in a ksfq buffer for processing metadata
	DATA_BUFFER_SIZE	NUMBER	number of blocks in a ksfq buffer for processing data
1217	DEGREE	NUMBER	Number of worker processes that can be active at any time
1219	ERROR_COUNT	NUMBER	Number of errors reported for job

job state row 1205A (517)

Fig. 12

1301

TOTAL_BYTES	NUMBER	For Export, estimated size of data in job. For Import, size of data in job to be loaded.
PLATFORM	VARCHAR2(100)	Platform used for the Export/Import
INSTANCE	VARCHAR2(15)	Instance name that job is running upon (RAC only)
ABORT_STEP	NUMBER	Process_order number of Object row that will cause the job to abort. For testing purposes only.
SCN	NUMBER	The System Change Number passed to Logical Standby for all DDL creations. This is *not* the SCN used for TABLE_DATA OBJECT rows; they each have their own.
OBJECT_TYPE_PATH	VARCHAR2(200)	Final termination message for job
OLD_VALUE	VARCHAR2(4000)	This is the opaque "cookie" returned to us by dbms_internal.safe_scn.next_scn during export and handed in during import & network. Used by Logical Standby / Streams
FLAGS	NUMBER	X00000001 Job is interesting for either streams or logical standby and the SCNs for tables need to be supplied via the safe_scn package.

job state row 1205B (517)

Column Name	Datatype	Meaning
PROCESS_ORDER	NUMBER	-3 for Export jobs.
DUPLICATE	NUMBER	0
SEED	NUMBER	Highest value of PROCESS_ORDER in Master Table. Originally set to 1 when Master Table is initially created.

MAX_PROCESS_ORDER row 1303 (517)

Column Name	Datatype	Meaning
PROCESS_ORDER	NUMBER	-5 for Export jobs. -6 for Import and SQL_File jobs.
DUPLICATE	NUMBER	The sequence number for each object type
OBJECT_TYPE_PATH	VARCHAR2(200)	Path of object type whose processing has completed.
OBJECT_PATH_SEQNO	NUMBER	Sequence number for object type. This field orders object types for import.
COMPLETION_TIME	DATE	Time when processing of object was completed.
COMPLETED_ROWS	NUMBER	The number of objects of current object path have been processed.

504

1307

TYPE_COMPLETION row 1305 (517)

Fig. 13

	Column Name	Datatype	Meaning
503	PROCESS_ORDER	NUMBER	-21 for Export jobs. -22 for Import and SQL_File jobs.
504	DUPLICATE	NUMBER	Internal number assigned to each file as it is specified at export time. File references by OBJECT rows always use this number rather than the file name.
1403	FILE_TYPE	NUMBER	0 if disk file.
1405	USER_DIRECTORY	VARCHAR2(4000)	Directory path used for dumpfile.
	USER_FILE_NAME	VARCHAR2(4000)	Original file name specified by user
	FILE_NAME	VARCHAR2(4000)	Fully resolved name (including path information) for file.
1407	FILE_MAX_SIZE	NUMBER	Maximum size for the file. 0 if file is extendable.

FILE row 1401 (521)

	Column Name	Datatype	Meaning
503	PROCESS_ORDER	NUMBER	-23 for Export jobs. -24 for Import and SQL_File jobs.
504	DUPLICATE	NUMBER	Ordinal position for when this ADD_FILE was specified (used to maintain round robin ordering between wildcarded names).
1411	FILE_NAME	VARCHAR2(4000)	Template for filename including substitution variables.
1413	FILE_MAX_SIZE	NUMBER	Maximum size for the file. 0 if file is extendable.
1415	SEED	NUMBER	Last value used for resolving substitution variables. Each wildcard specification needs a unique number since ADD_FILEs can come in after the job has started.
1417	LAST_FILE	NUMBER	File number of last file resolved from this wildcard string. This column is used to identify where we are in the round robin expansion of wildcarded names.

WILDCARD_FILE row 1409 (521)

Fig. 14

	Column Name	Datatype	Meaning
503	PROCESS_ORDER	NUMBER	-41 for Export jobs. -42 for Import and SQL_File jobs.
504	DUPLICATE	NUMBER	Internal Id for the worker process.
1503	PROCESS_NAME	VARCHAR2(30)	Process name for the worker process
1505	OBJECT_NUMBER	NUMBER	PROCESS_ORDER value for the object being processed by the worker process.
	OBJECT_SCHEMA	VARCHAR2(30)	The schema of the object being processed. Null if not in EXECUTING state or processing a non-schema object.
	OBJECT_NAME	VARCHAR2(500)	The name of the object being processed. Null if not in EXECUTING state or processing an unnamed object.
	OBJECT_LONG_NAME	VARCHAR2(4000)	The name of the object being processed. Null if not in EXECUTING state or processing an unnamed object.
	OBJECT_TYPE_PATH	VARCHAR2(200)	The object type pathname of the object being processed. Null if not in EXECUTING state.
	PARTITION_NAME	VARCHAR2(30)	The name of the partition of the object being processed. Only
			object within a partitioned table; Null otherwise.
1507	TOTAL_BYTES	NUMBER	Number of bytes within a TABLE_DATA object. On EXPORT, number may be a estimate. NULL if no estimate is available on object or if work is not processing a TABLE_DATA object.
	COMPLETED_ROWS	NUMBER	Number of data rows written or read for current TABLE_DATA object. For other objects, the number of objects of current object path have been processed.
	LAST_UPDATE	DATE	Time of last update for Worker row. Used to approximate the time of a crash during restart.
	WORK_ITEM	VARCHAR2(30)	Current work item being processed by Worker. NULL if worker is idle. Possible values are UNLOAD METADATA, UNLOAD DATA, LOAD METADATA, LOAD DATA, ESTIMATE JOB, SQL FILE JOB, RELEASE FILES, and EXITING
	STATE	VARCHAR2(30)	One of the following values: WORK_WAITING, FILE_WAITING, EXECUTING, MASTER. WORK_WAITING -- worker is waiting for work from the Master Control Process. FILE_WAITING -- worker is waiting for a file space from the Master Control Process. EXECUTING -- worker is processing one or more objects. See OBJECT_NUMBER and OBJECT_ROWID columns for details. MASTER -- Worker Process is either saving or restoring the Master Table to/from the dump file set.
1509	METADATA_IO	NUMBER	Amount of Metadata written to the dump file (for export) or read from the dump file (for Import) or transferred over the link since last restart for this Worker.
	DATA_IO	NUMBER	Amount of table data written to the dump file (for Export) or read from the dump file (for Import) or transferred over the link since last restart for this Worker.
	CUMULATIVE_TIME	NUMBER	The amount of time that this worker process has spent actively processing the job.

WORKER row 1501 (517)

Fig. 15

503

504

1603

Column Name	Datatype	Meaning
PROCESS_ORDER	NUMBER	-7 for Export jobs. -8 for Import and SQL_File jobs.
DUPLICATE	NUMBER	0
METADATA_IO	NUMBER	Amount of Metadata written to the dump file (for export) or read from the dump file (for Import) or transferred over the link (for Import over a network).
DATA_IO	NUMBER	Amount of table data written to the dump file (for Export) or read from the dump file (for Import) or transferred over the link (for Import over a network).
TOTAL_BYTES	NUMBER	An estimate of the total size of the job. For Import from files, the total size of the requested data within the dump file.
CUMULATIVE_TIME	NUMBER	Sum of the amount of time that each worker process has spent actively processing the job in hundredths of a second. If 3 workers were active for an hour, this column would contain 1080000.

RESTART_STATUS row 1601 (517)

503

504

1605

1606

1607

Column Name	Datatype	Meaning
PROCESS_ORDER	NUMBER	-9 for Export jobs. -10 for Import jobs.
DUPLICATE	NUMBER	Unique key assigned to each (restart of the job. The first start of a job will be represented by DUPLICATE 0.
METADATA_IO	NUMBER	Bytes of Metadata written to the dump file (for export) or read from the dump file (for Import) or transferred over the link (for Network).
DATA_IO	NUMBER	Bytes of table data written to the dump file (for Export) or read from the dump file (for Import) or transferred over the link (for Network).
TOTAL_BYTES	NUMBER	For Export and Network, an estimate of the total size of the Operation (if available). For Import, the total size of the requested data within the dump file.
ERROR_COUNT	NUMBER	Number of errors reported for job
CUMULATIVE_TIME	NUMBER	Sum of the amount of time that each worker process has spent actively processing the job. If 3 workers were active for an hour, this column would contain 1080000. Using DATA_IO, TOTAL_BYTES, CUMULATIVE_TIME, and JOB_STATE.DEGREE, an estimate of the remaining time for the job will be possible.
OBJECT_TYPE_PATH	VARCHAR2(200)	Final termination message from previous job incarnation.
ELAPSED_TIME	NUMBER	Amount of time that elapsed between the restart and the latest timestamp found on the next restart in 100ths of seconds
START_TIME	DATE	Starting time for previous incarnation of job.
PLATFORM	VARCHAR2(100)	Platform used during previous incarnation of job.
INSTANCE	VARCHAR2(15)	Instance name that job incarnation ran upon (RAC only)
DEGREE	NUMBER	Degree of parallelism at end of previous incarnation of job.

RESTART row 1605 (519)

Fig. 16

	Column Name	Datatype	Meaning
503	PROCESS_ORDER	NUMBER	-51 for Export jobs. -52 for Import and SQL_File jobs.
504	DUPLICATE	NUMBER	Internal Id for distinguishing Data Filters
1703	NAME	VARCHAR2(30)	Name of filter.
1705	VALUE_T	VARCHAR2(4000)	Definition of a text filter.
1707	VALUE_N	NUMBER	Definition of a numerical filter.
	OBJECT_SCHEMA	VARCHAR2(30)	Schema of table to which filter applies
	OBJECT_NAME	VARCHAR2(30)	Table for which filter applies

DATA_FILTER row 1701 (523)

	Column Name	Datatype	Meaning
503	PROCESS_ORDER	NUMBER	-53 for Export jobs. -54 for Import and SQL_File jobs.
504	DUPLICATE	NUMBER	Internal Id for distinguishing Metadata Filters
1711	NAME	VARCHAR2(30)	Name of filter.
1713	VALUE_T	VARCHAR2(4000)	Definition of filter
	OBJECT_TYPE_PATH	VARCHAR2(200)	Object class affected by the filter. If NULL, the filter affects all object classes.

METADATA_FILTER row 1709 (523)

	Column Name	Datatype	Meaning
503	PROCESS_ORDER	NUMBER	-57 for Export and Estimate jobs -58 for Import, Network and SQL_File jobs.
504	DUPLICATE	NUMBER	Internal Id for distinguishing Metadata transforms
1717	NAME	VARCHAR2(30)	Name of Remap or Transform. Legal name is: SEGMENT_ATTRIBUTES.
1719	OLD_VALUE	VARCHAR2(4000)	Specifies value to be remapped for remaps. Null otherwise.
1721	VALUE_T	VARCHAR2(4000)	Specifies new value for remaps. For transforms, specifies the value.
1723	VALUE_N	NUMBER	Definition of a numerical filter.
	OBJECT_TYPE	VARCHAR2(30)	Object class affected by the remap or transform. If NULL, the remap or transform affects all applicable object classes.

METADATA_TRANSFORM row 1715 (525)

Fig. 17

	Column Name	Datatype	Meaning
503	PROCESS_ORDER	NUMBER	-59 for Export jobs. -60 for Import and SQL_File jobs.
504	DUPLICATE	NUMBER	Internal Id for distinguishing Parameters
1803	NAME	VARCHAR2(30)	Name of PARAMETER.
	IS_DEFAULT	NUMBER	If non-zero, parameter setting was not supplied by the client.
	VALUE_T	VARCHAR2(4000)	Specifies the value set for a text parameter.
	VALUE_N	NUMBER	Definition of a numerical parameter.

PARAMETER row 1801 (527)

	Column Name	Datatype	Meaning
503	PROCESS_ORDER	NUMBER	-73 for Export jobs. -74 for Import and SQL_File jobs.
504	DUPLICATE	NUMBER	0
1807	VALUE_T	VARCHAR2(4000)	A DDL command to reestablish the NLS settings for the job.

NLS_PARAMS row 1805 (527)

Fig. 18

```
DECLARE
      handle      NUMBER;

BEGIN
1903      handle := DBMS_DATAPUMP.OPEN( 'EXPORT', 'FULL', NULL,
                                        'MYDBMOVE_EXPORT');
          DBMS_DATAPUMP.ADD_FILE(handle, 'file1.dmp',
                                 'MY_DIR1', '600M');
1905      DBMS_DATAPUMP.ADD_FILE(handle, 'file2.dmp',
                                 'MY_DIR2', '600M');
          DBMS_DATAPUMP.ADD_FILE(handle, 'file3.dmp',
                                 'MY_DIR3', '600M');
1907      DBMS_DATAPUMP.METADATA_FILTER(handle, 'SCHEMA_EXPR',
                                        '!= ''BLAKE''');
1909      DBMS_DATAPUMP.SET_PARALLEL(handle, 3);
1911      DBMS_DATAPUMP.START_JOB(handle);
1913      DBMS_DATAPUMP.DETACH(handle);
END;
```

1901

```
DECLARE
   handle      NUMBER;

BEGIN
1917   handle := DBMS_DATAPUMP.ATTACH ('MYDBMOVE_EXPORT');
1919   DBMS_DATAPUMP.STOP_JOB(handle, 1, 1, 0);
END;
```

1915

```
DECLARE
   handle      NUMBER;

BEGIN
1923  handle := DBMS_DATAPUMP.ATTACH ('MYDBMOVE_EXPORT');
      DBMS_DATAPUMP.ADD_FILE(handle, 'file4.dmp',
1925                         'MY_DIR4', '600M');
      DBMS_DATAPUMP.ADD_FILE(handle, 'file5.dmp',
                             'MY_DIR5', '600M');

1927   DBMS_DATAPUMP.SET_PARALLEL(handle, 5);
1929   DBMS_DATAPUMP.START_JOB(handle);
1931   DBMS_DATAPUMP.DETACH(handle);
END;
```

1921

Fig. 19

```
BEGIN
  2003 handle := DBMS_DATAPUMP.OPEN( 'IMPORT','FULL', NULL,
                                     'MYDBMOVE_IMP');

  2005 DBMS_DATAPUMP.SET_PARAMETER(handle, 'KEEP_MASTER', 0);
       DBMS_DATAPUMP.ADD_FILE(handle, 'file1.dmp',
                                     'MY_NEWDIR1', '600M');
       DBMS_DATAPUMP.ADD_FILE(handle, 'file2.dmp',
                                     'MY_NEWDIR2', '600M');
2007   DBMS_DATAPUMP.ADD_FILE(handle, 'file3.dmp',
                                     'MY_NEWDIR3', '600M');
       DBMS_DATAPUMP.ADD_FILE(handle, 'file4.dmp',
                                     'MY_NEWDIR4', '600M');
       DBMS_DATAPUMP.ADD_FILE(handle, 'file5.dmp',
                                     'MY_NEWDIR5', '600M');

  2009 DBMS_DATAPUMP.METADATA_REMAP(handle, 'MAP_TABLESPACE',
                                    'USER1', 'NEWUSER1');
  2011 DBMS_DATAPUMP.SET_PARALLEL(handle, 4);
  2013 DBMS_DATAPUMP.START_JOB(handle);
  2015 DBMS_DATAPUMP.DETACH(handle);
END;
```

2001

Fig. 20

Table 1: APIs and Job states

API	Valid Job states for API	Description
ADD_FILE 2103	Defining Executing[a] Idling[a] Stop pending	Specifies a file for the dump file set, or the location of the log file or the location of the file to receive the SQL_FILE output.
ATTACH 2105	Defining Executing Idling Stop pending Stopped Completing Completed	Allows a user session to monitor a job
DATA_FILTER 2107	Defining	Restricts data processed by a job
DETACH 2109	Defining Executing Idling Stop pending Completing	Disconnects a user session from a job
GET_STATUS 2111		Obtains the status of a job
LOG_ENTRY 2113		Adds an entry to the log file
METADATA_FILTER 2115	Defining	Restricts metadata processed by a job
METADATA_REMAP 2117	Defining	Remaps metadata processed by a job
METADATA_TRANSFORM 2119	Defining	Alters metadata processed by a job
OPEN 2121	Undefined	Creates a new job
SET_PARALLEL 2123	Defining Executing Idling Stop pending	Specifies parallelism for a job
SET_PARAMETER 2125	Defining	Alters default processing by a job
START_JOB 2127	Defining Idling	Begins/resumes executing a job
STOP_JOB 2129	Defining Executing Idling Stop pending	Initiates orderly shutdown of a job

a. Export jobs only

Fig. 21

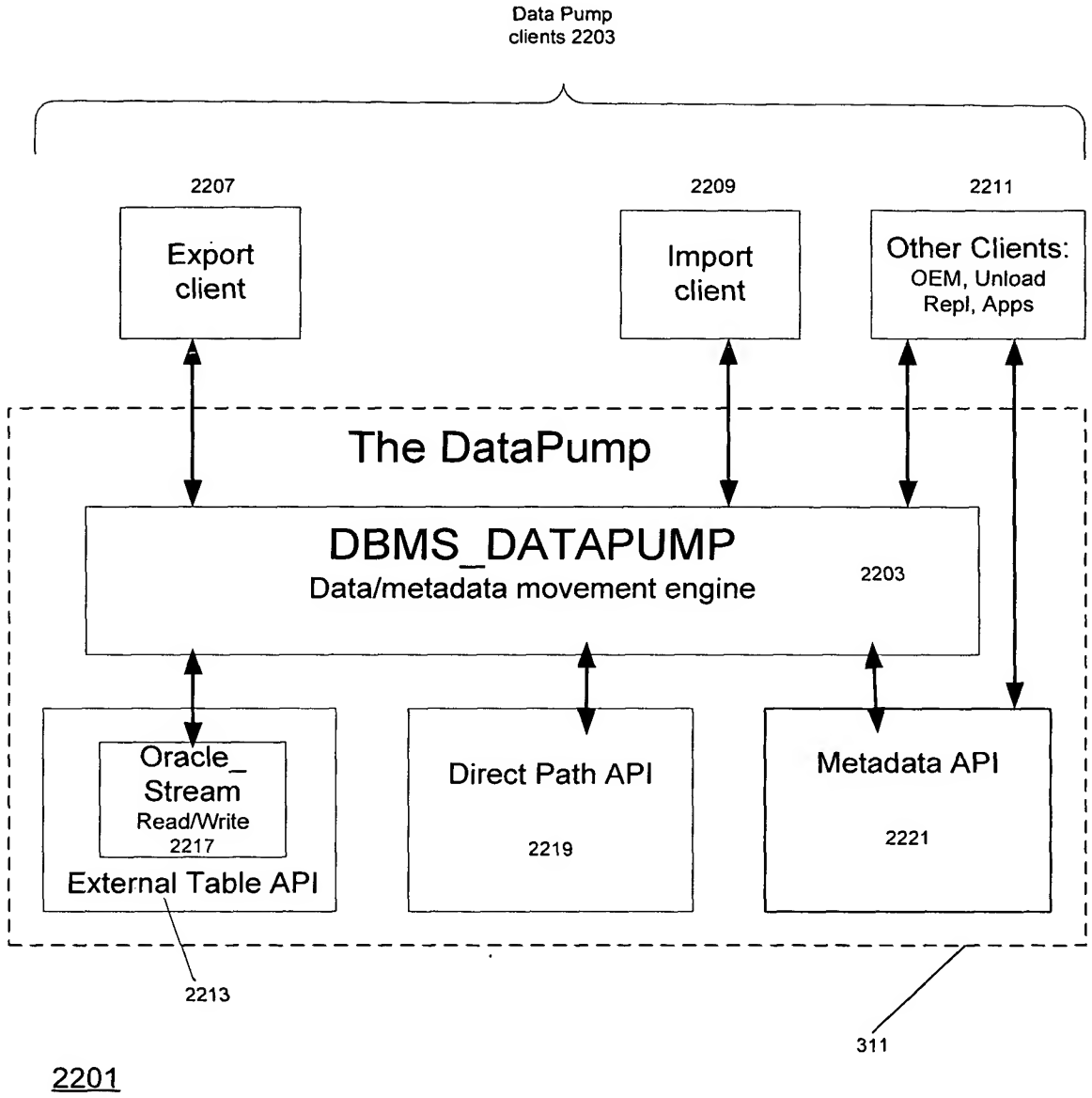

Data Pump
clients 2203
2207
2209
2211
Export
client
Import
client
Other Clients:
OEM, Unload
Repl, Apps
The DataPump
DBMS_DATAPUMP
Data/metadata movement engine
2203
Oracle_
Stream
Read/Write
2217
External Table API
Direct Path API
2219
Metadata API
2221
2213
311
2201

Fig. 22

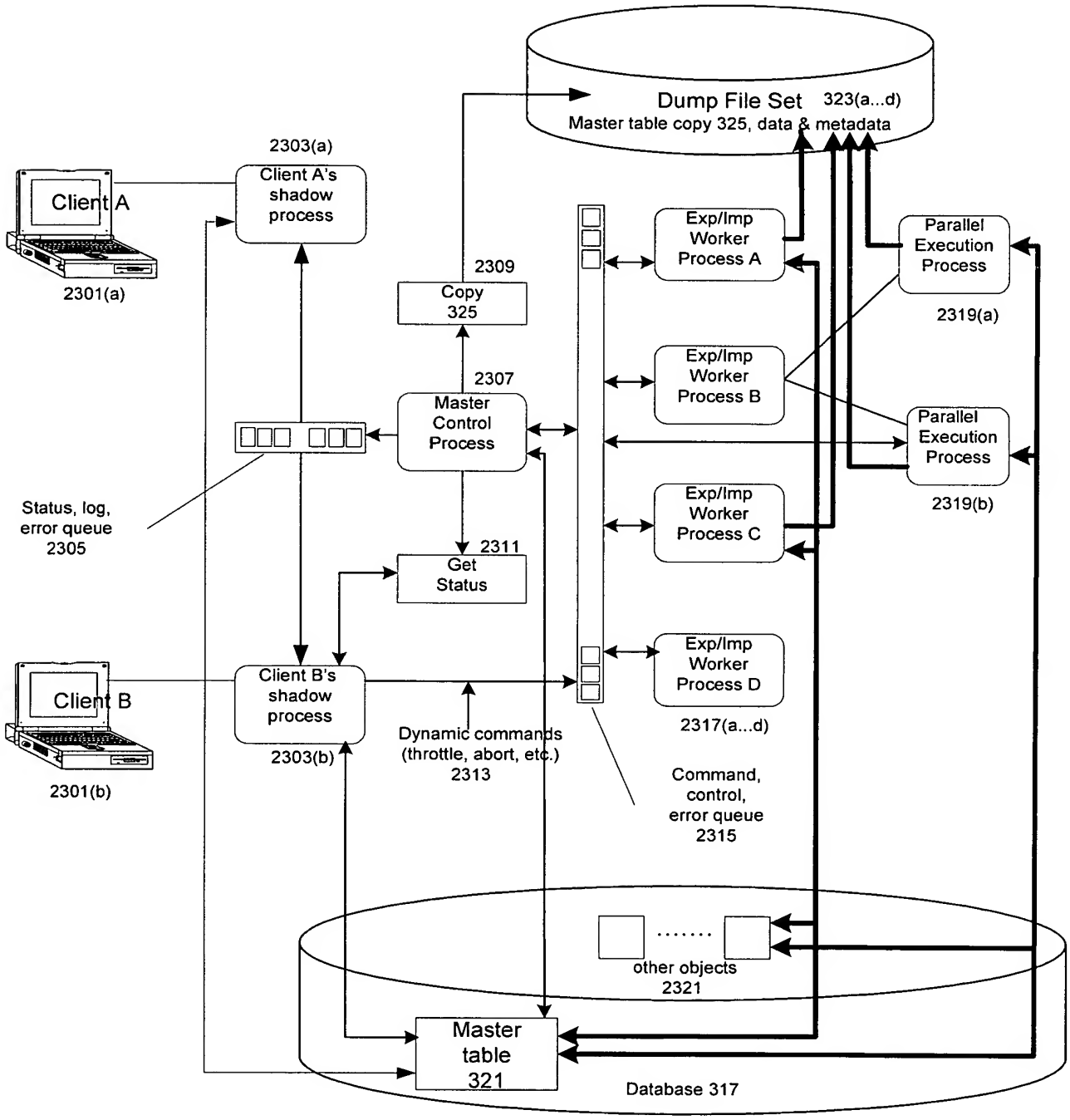

Dump File Set 323(a...d)
Master table copy 325, data & metadata
2303(a)
Client A's shadow process
Client A
2301(a)
2309
Copy 325
2307
Master Control Process
Exp/Imp Worker Process A
Exp/Imp Worker Process B
Exp/Imp Worker Process C
Exp/Imp Worker Process D
Parallel Execution Process
2319(a)
Parallel Execution Process
2319(b)
2317(a...d)
Status, log, error queue 2305
2311
Get Status
Client B's shadow process
2303(b)
Client B
2301(b)
Dynamic commands (throttle, abort, etc.) 2313
Command, control, error queue 2315
other objects 2321
Master table 321
Database 317

Fig. 23

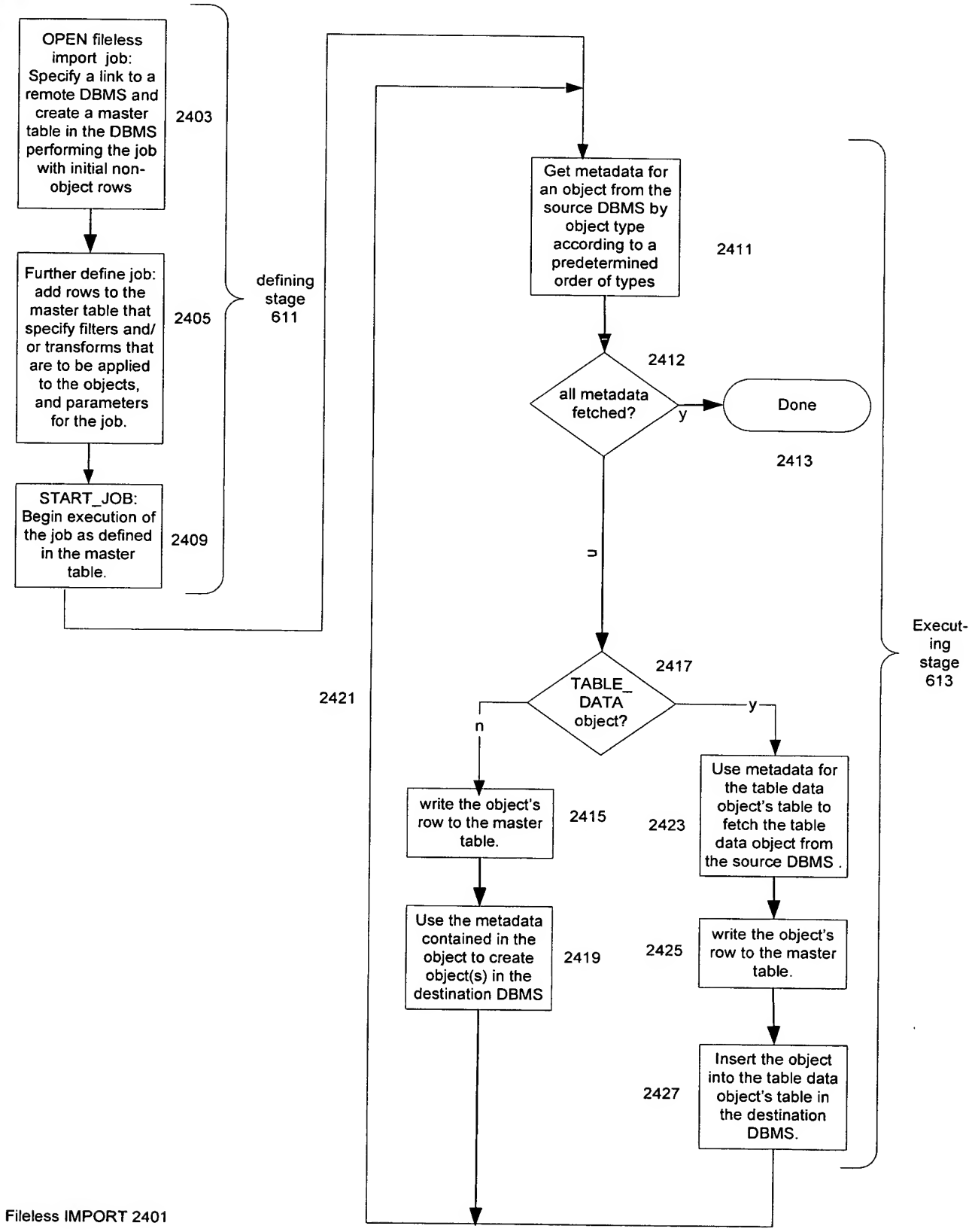

OPEN fileless import job: Specify a link to a remote DBMS and create a master table in the DBMS performing the job with initial non-object rows
2403
Further define job: add rows to the master table that specify filters and/ or transforms that are to be applied to the objects, and parameters for the job.
2405
START_JOB: Begin execution of the job as defined in the master table.
2409
defining stage 611
Get metadata for an object from the source DBMS by object type according to a predetermined order of types
2411
2412
all metadata fetched?
y
Done
2413
n
2417
TABLE_ DATA object?
n
y
2421
write the object's row to the master table.
2415
Use the metadata contained in the object to create object(s) in the destination DBMS
2419
2423
Use metadata for the table data object's table to fetch the table data object from the source DBMS .
2425
write the object's row to the master table.
2427
Insert the object into the table data object's table in the destination DBMS.
Executing stage 613
Fileless IMPORT 2401

Fig. 24